PART IV

Composite Types

Gopher Level

Intermediate

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

PART IV

Composite Types

✓

Arrays

Collection
of
Elements
Indexable
Fixed Length

✓

Slices

Collection
of
Elements
Indexable
Dynamic length

Strings Revisited

Read-Only Byte Slices
Indexable
Fixed Length

Maps

Collection
of
Indexable
Key-Value Pairs

Structs

Groups
different types of
variables together

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

PART IV

Composite Types

✓

Arrays

Collection
of
Elements
Indexable
Fixed Length

✓

Slices

Collection
of
Elements
Indexable
Dynamic length

Strings Revisited

Read-Only Byte Slices
Indexable
Fixed Length

Maps

Collection
of
Indexable
Key-Value Pairs

Structs

Groups
different types of
variables together

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

STRINGS REVISITED

What you're going to learn in this section?

- ★ Bytes, Runes, and Strings
- ★ Unicode Code Points
- ★ Character Sets
- ★ Printing and Converting
- ★ Indexing and Slicing Strings
- ★ String Internals

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

SPAM MASKER ***

"Click http://hi.com"

"Click http://******"

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

TEXT WRAPPER

This is an unwrapped text but it wants to be wrapped.

This is an unwrapped
text but it wants
to be wrapped.

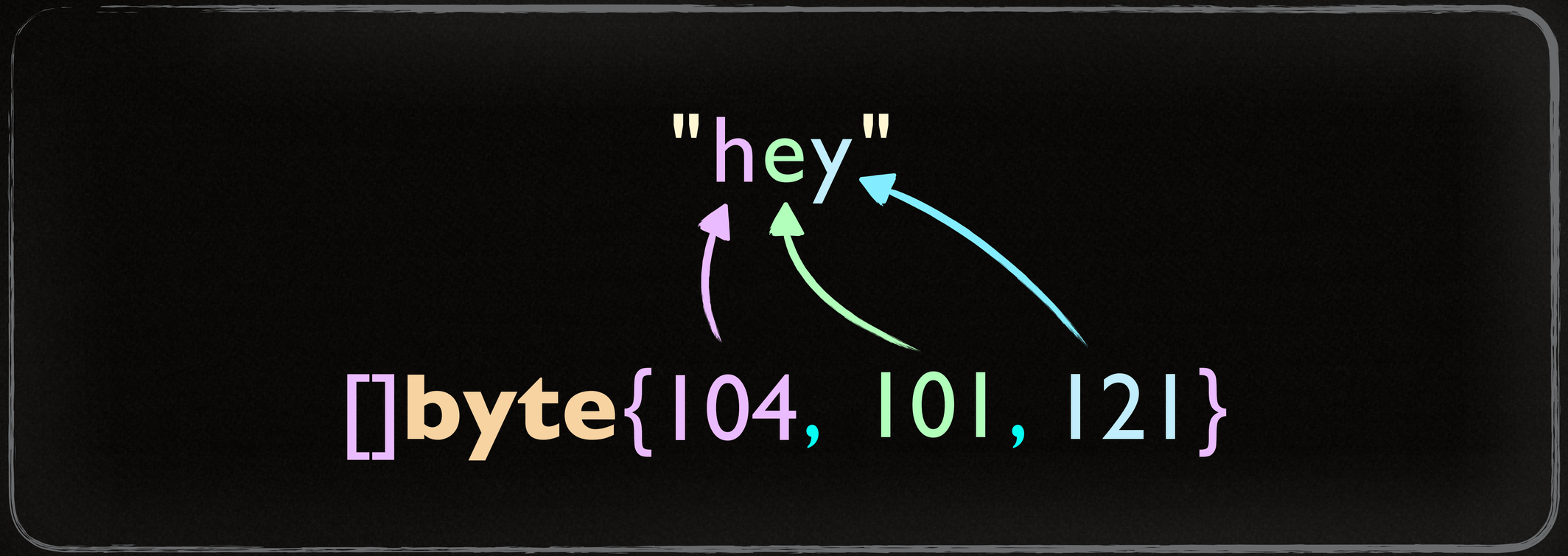
Copyright 2019 Inanc Gumus — Twitter: @inancgumus

Bytes, Runes, and Strings

The Basics

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

A string is just a series of bytes



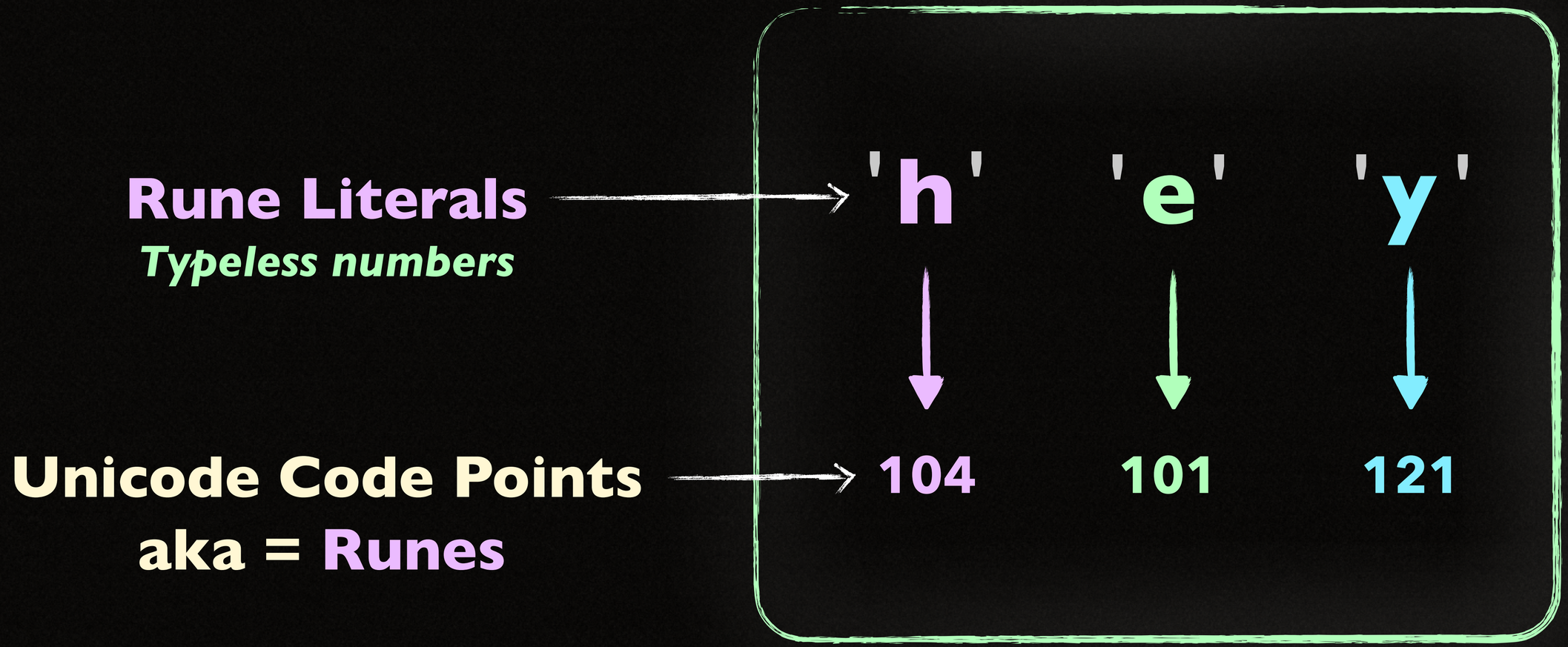
Copyright 2019 Inanc Gumus — Twitter: @inancgumus

string & []byte are interchangeably convertible

```
[]byte("hey")

string([]byte{104, 101, 121})
```

Copyright 2019 Inanc Gumus — Twitter: @inancgumus



A rune literal can represent:
byte (uint8), rune (int32), or any other integer type

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

'h'

Decimal

base 10

104

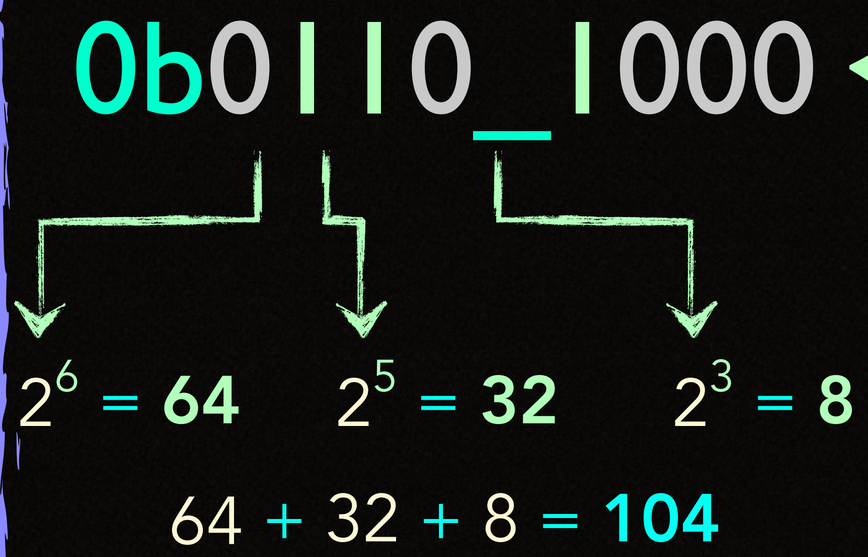
used by earthlings 🤷‍♂️🤷‍♀️

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

used by computers 🤖

Binary

base 2



Decimal

base 10

104

used by earthlings 🙋‍♂️🙋‍♀️

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

used by computers 🤖

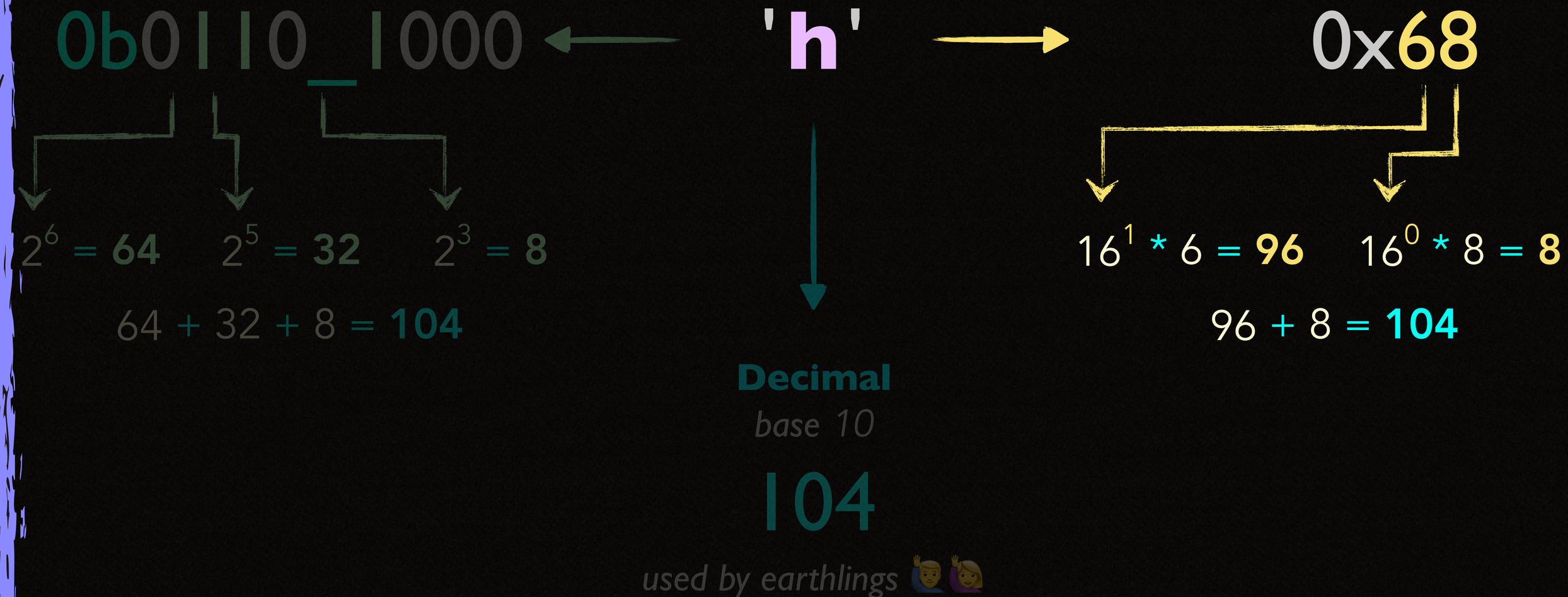
Binary

base 2

used by hackers 👩‍💻👨‍💻

Hexadecimal

base 16



Decimal

base 10

104

used by earthlings 🤷‍♂️🤷‍♀️

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

used by computers 🤖

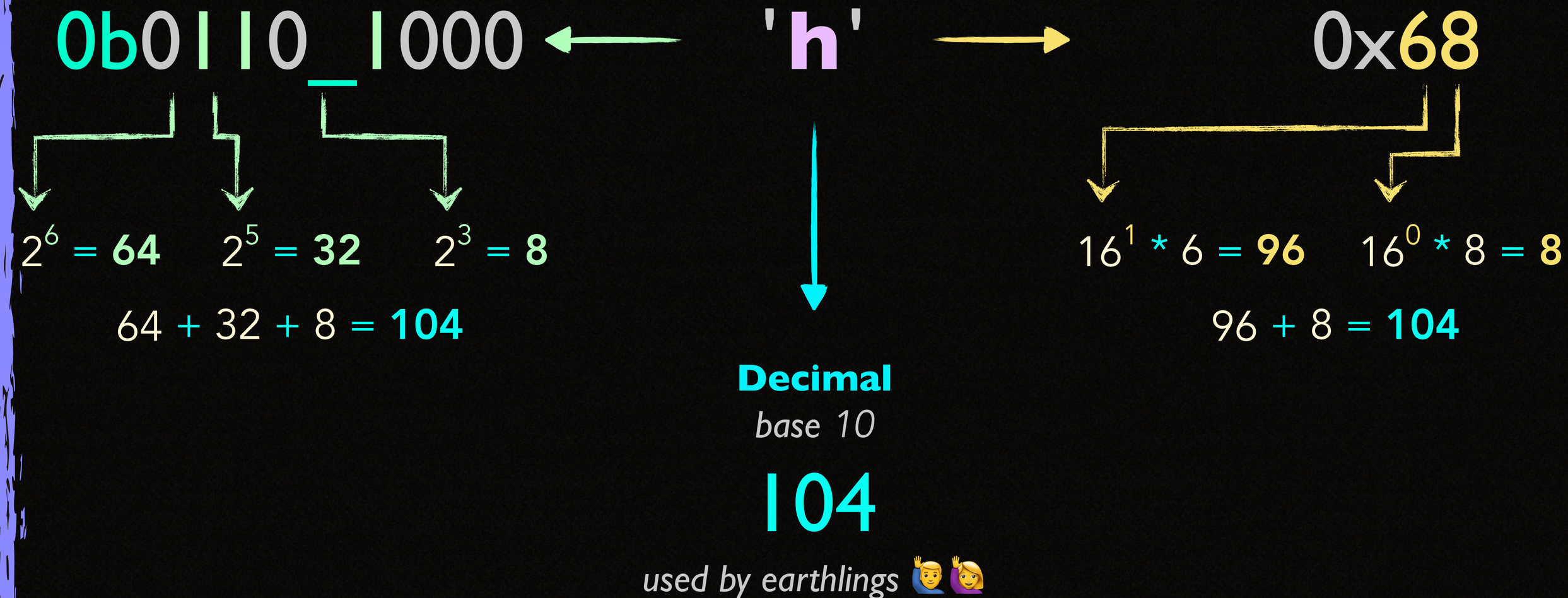
Binary

base 2

used by hackers 👩‍💻👨‍💻

Hexadecimal

base 16



Decimal

base 10

104

used by earthlings 🙋‍♂️🙋‍♀️

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

Bytes, Runes, and Strings

Let's write a character-set program

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

Bytes, Runes, and Strings

Converting, Indexing, and Slicing

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

string literals are automatically encoded in utf-8

the other string values do not need to be encoded in utf-8

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

Rune Decoding

How to get runes from an utf-8 encoded string without using `for range`?

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

DecodeRuneInString(" öykü ")

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

DecodeRuneInString(" öykü ")

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

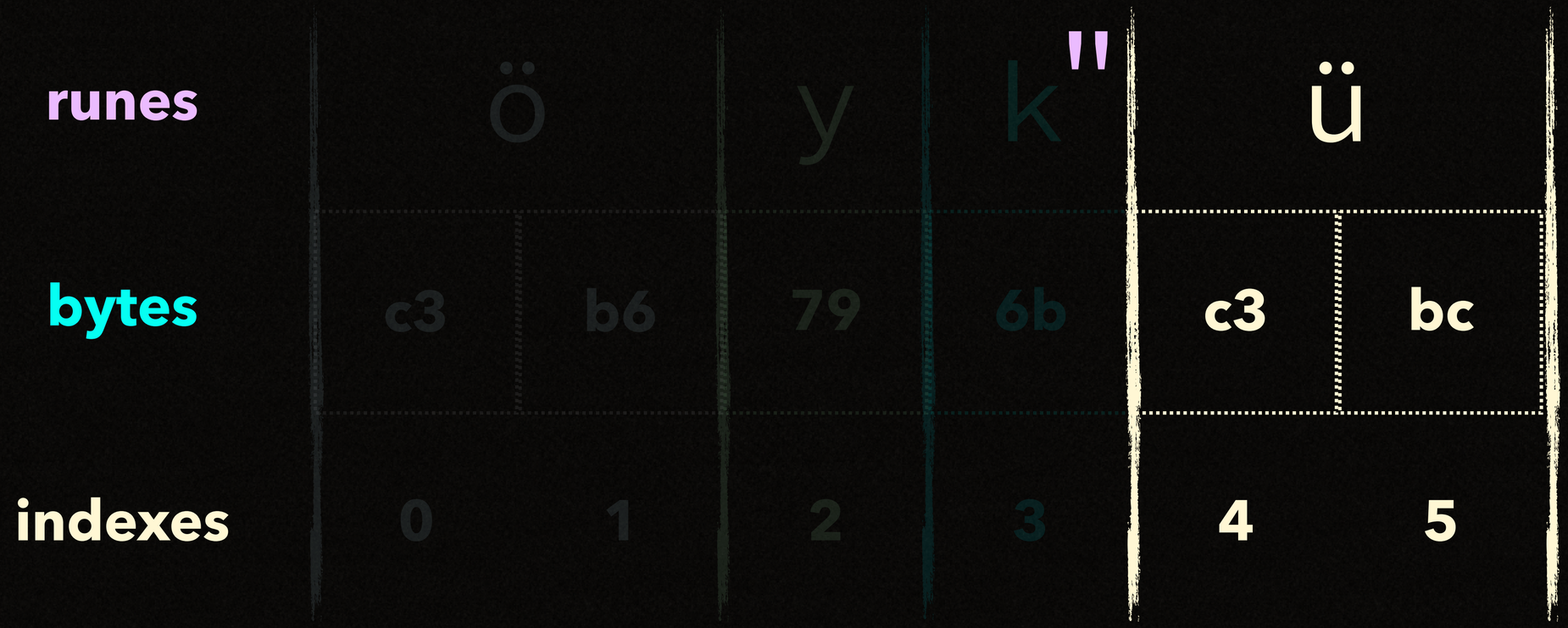
DecodeRuneInString("ykü")

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

DecodeRuneInString("kü")

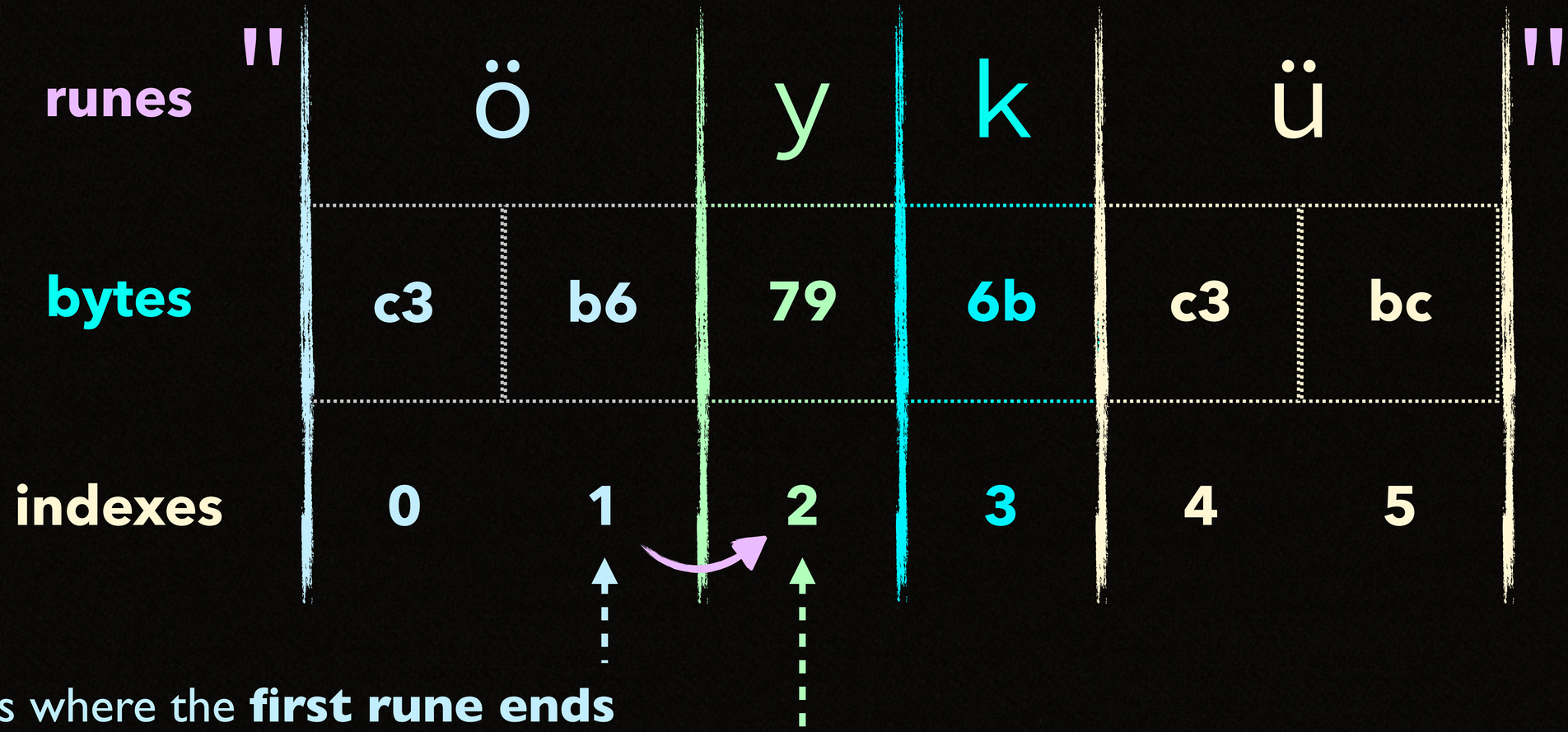
Copyright 2019 Inanc Gumus — Twitter: @inancgumus

DecodeRuneInString("ü")



Copyright 2019 Inanc Gumus — Twitter: @inancgumus

DecodeRuneInString(" öykü " [2:])



Copyright 2019 Inanc Gumus — Twitter: @inancgumus

String Internals

Why string values are read-only (immutable)?

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

STRING INTERNALS

A string is a data structure that points to a read-only backing array

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

STRING INTERNALS

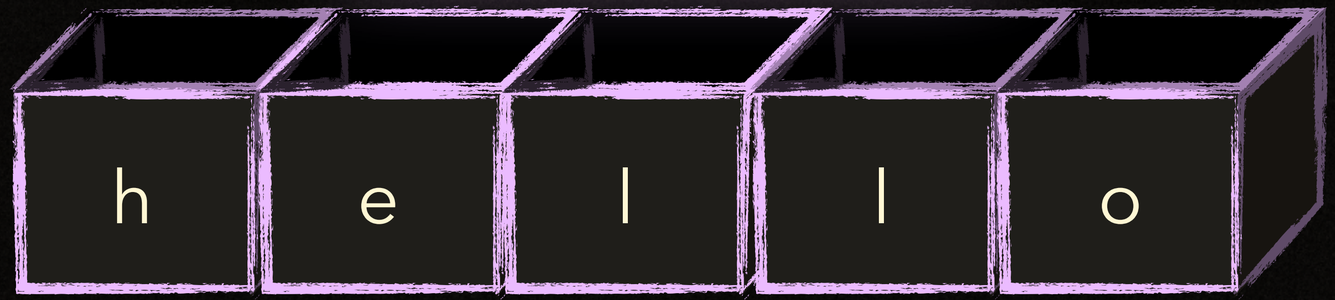
A string is a data structure that points to a read-only backing array

STRING HEADER

Pointer	57
Length	5
Capacity	5

→ **len("hello") = 5**

BACKING ARRAY



Copyright 2019 Inanc Gumus — Twitter: @inancgumus

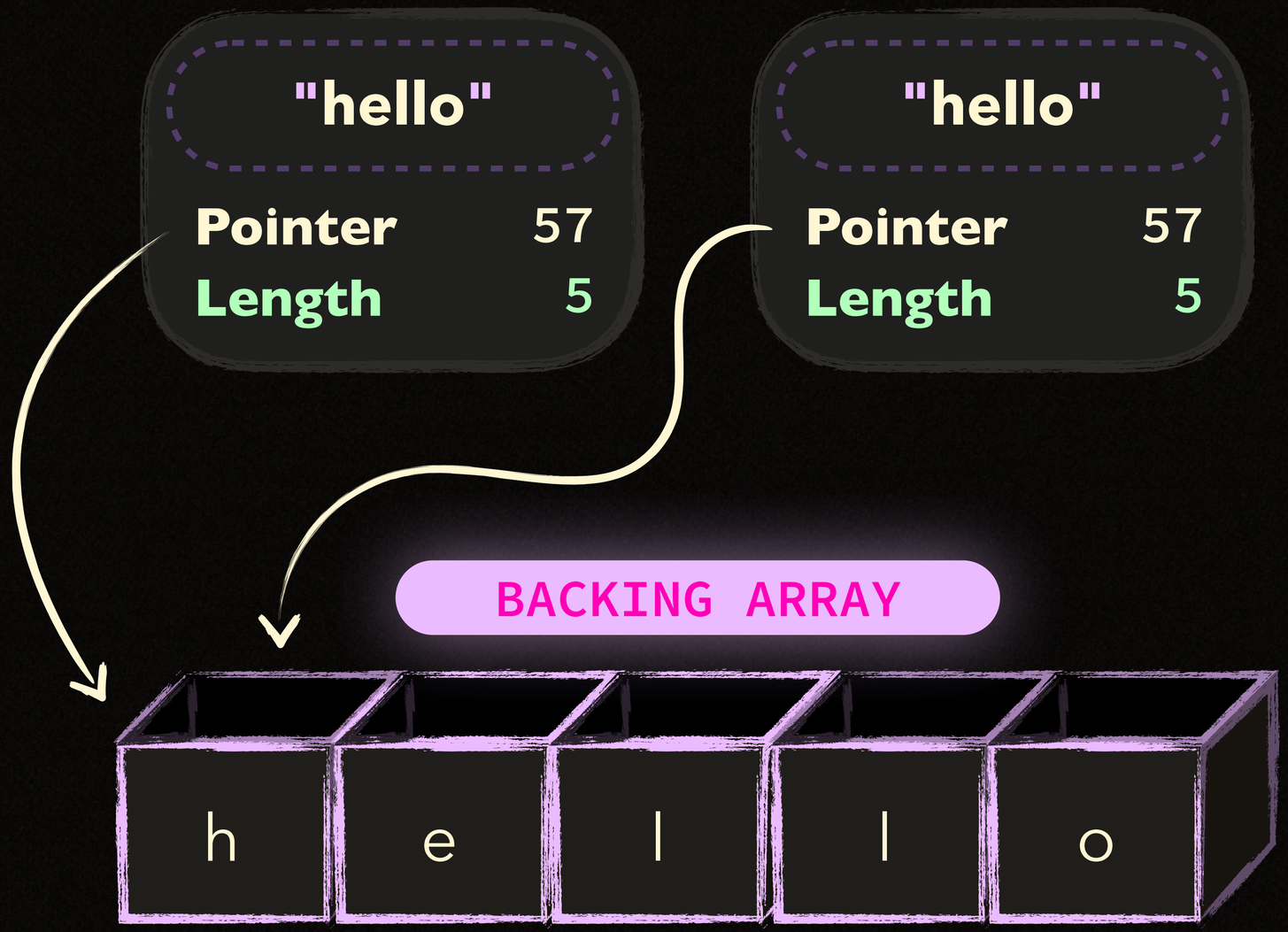
STRING INTERNALS

String Header stores where a string value starts and ends in computer memory

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

STRING INTERNALS

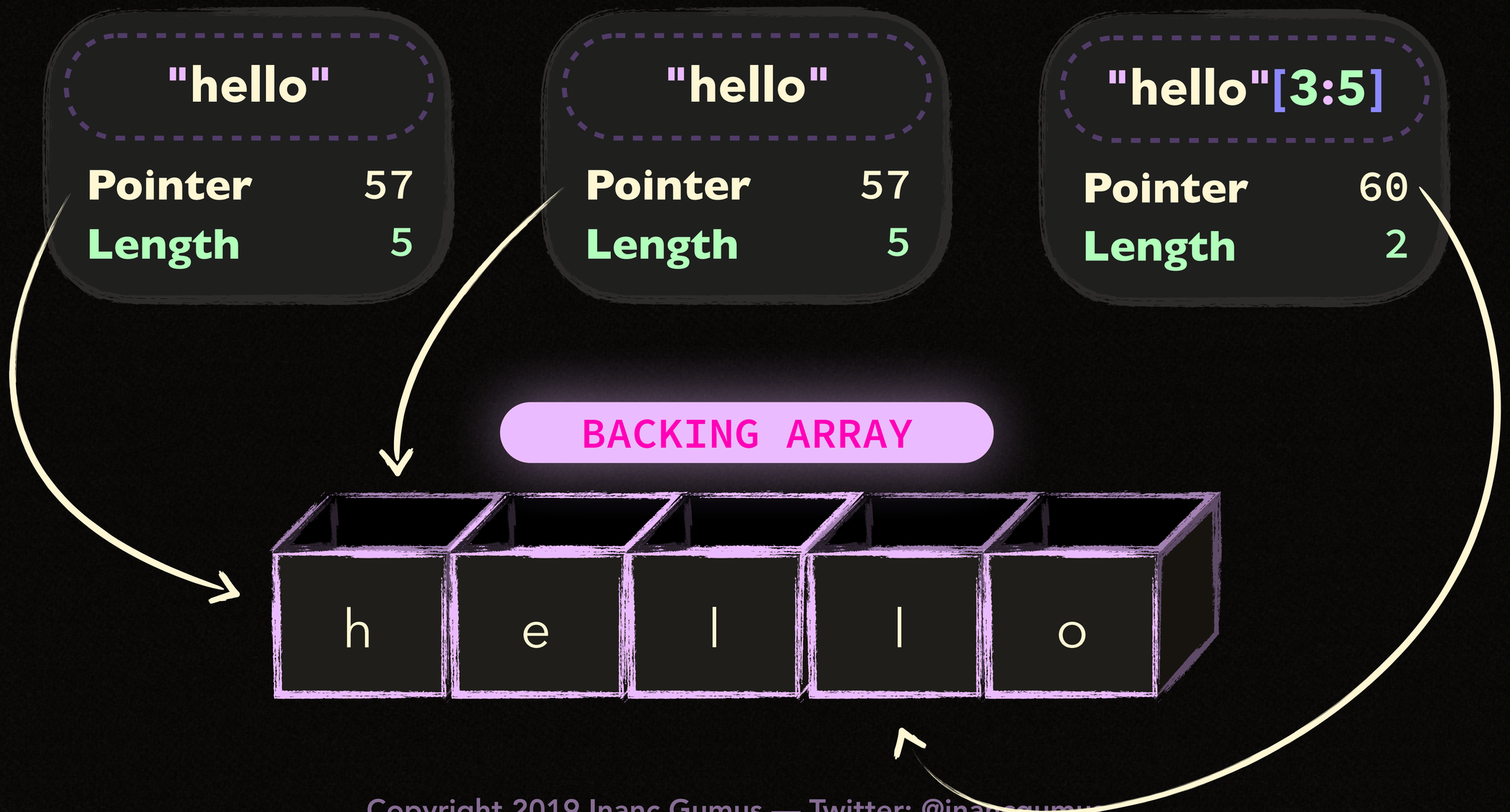
The backing array is shared among the same string values



Copyright 2019 Inanc Gumus — Twitter: @inancgumus

STRING INTERNALS

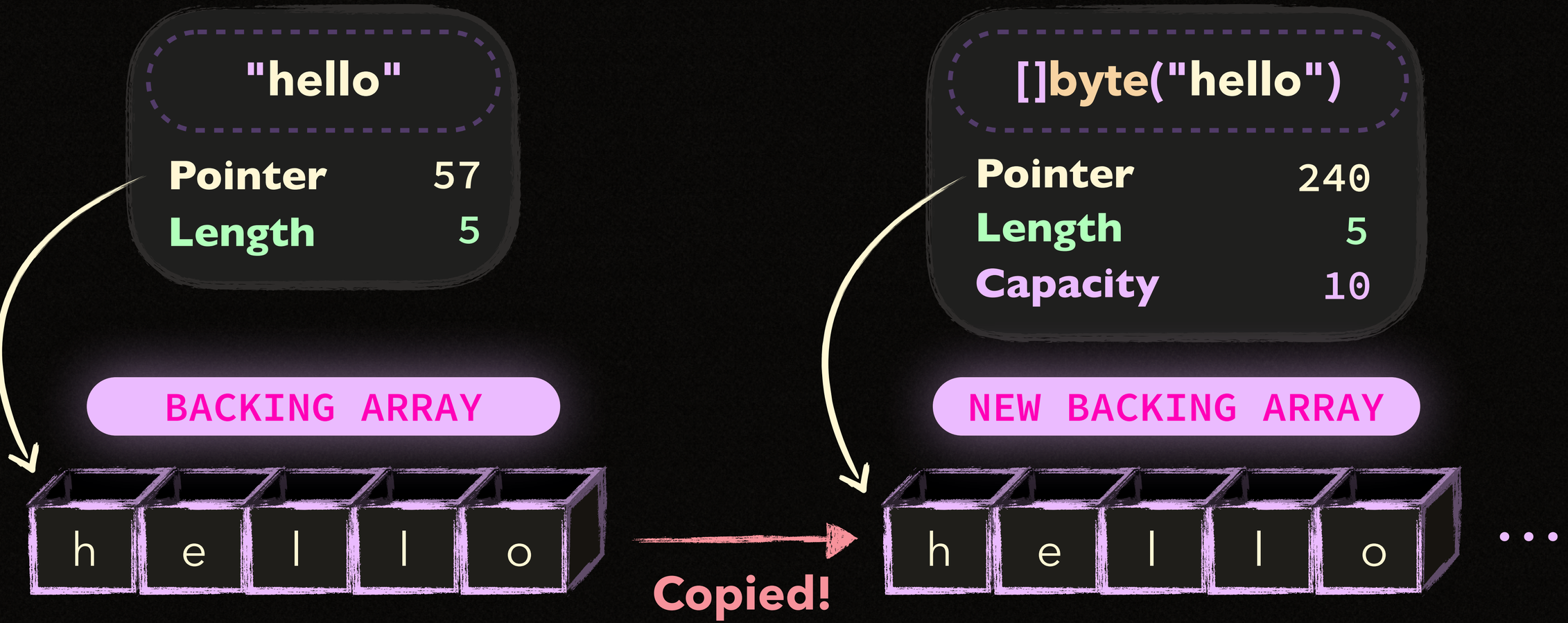
The backing array is also shared among the sliced string values



Copyright 2019 Inanc Gumus — Twitter: @inancgumus

STRING INTERNALS

Conversion may allocate a new backing array!



Copyright 2019 Inanc Gumus — Twitter: @inancgumus

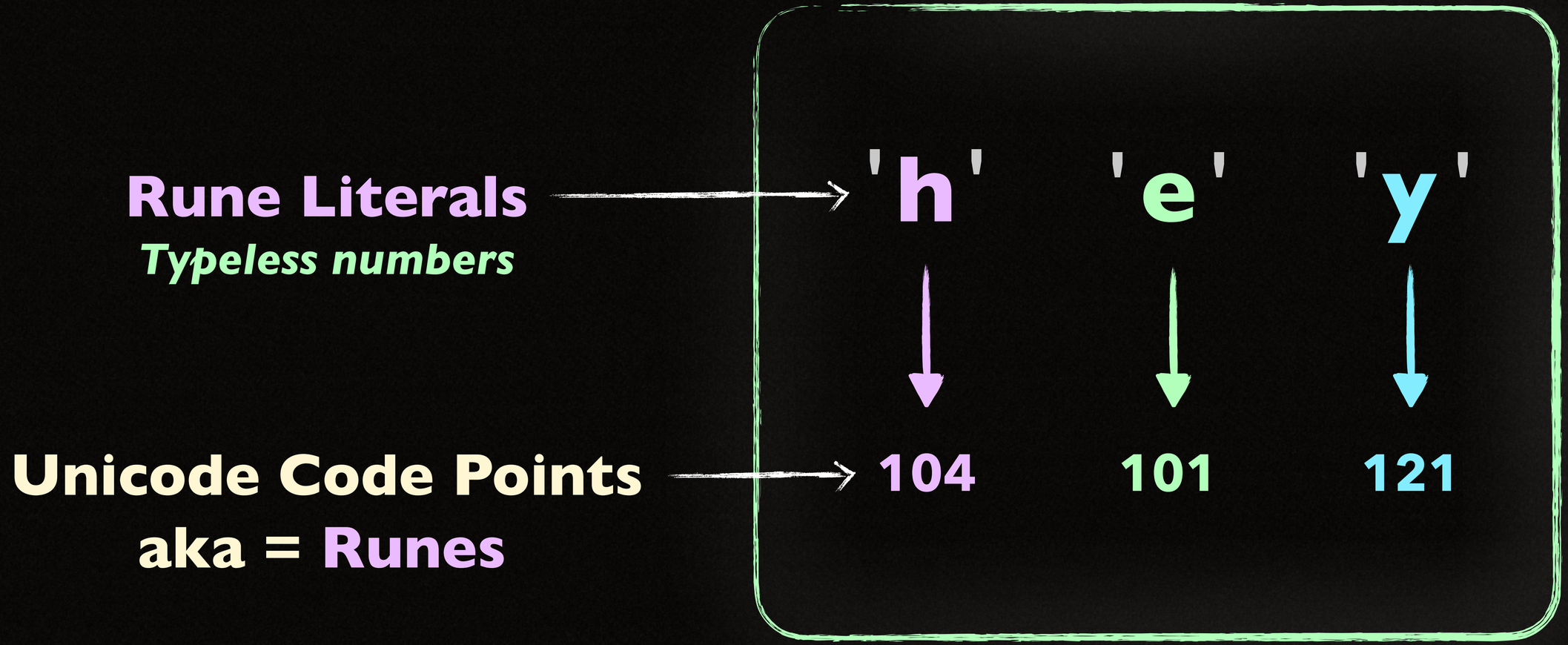
summary

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

```
[]byte("hey")

string([]byte{104, 101, 121})
```

Copyright 2019 Inanc Gumus — Twitter: @inancgumus



A rune literal can represent:
A byte, a rune, or any other numeric type

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

- 👉 **%s** prints []**byte** as **string**
- 👉 **%c** prints a rune as a character
- 👉 **%x** prints the bytes of a string in hexadecimal notation
- 👉 **% x** like **%x** but prints the bytes with spaces
- 👉 for range jumps over the runes in a string
- 👉 aString[index] returns a **byte**
- 👉 aString[start:stop] returns a **string**
- 👉 rune(aString) returns a **rune** slice by putting the runes inside the string
- 👉 **string**(anInteger) returns a string by utf-8 encoding the integer
- 👉 byteSlice = append(byteSlice, aString...)

  You can append a string to a byte slice
- 👉 utf8 and unicode packages contain helper functions for working with runes

Copyright 2019 Inanc Gumus — Twitter: @inancgumus

PART IV

Composite Types

✓

Arrays

Collection
of
Elements
Indexable
Fixed Length

✓

Slices

Collection
of
Elements
Indexable
Dynamic length

✓

Strings Revisited

Read-Only Byte Slices
Indexable
Fixed Length

Maps

Collection
of
Indexable
Key-Value Pairs

Structs

Groups
different types of
variables together

Copyright 2019 Inanc Gumus — Twitter: @inancgumus